Coupling distributed and symbolic execution for natural language queries

Lili Mou Zhengdong Lu Hang Li Zhi Jin

北京大学 PEKING UNIVERSITY
DEEPLY CURIOUS 深度好奇
HUAWEI

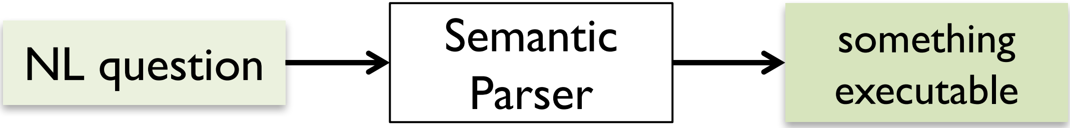
Outline

- Learning the semantics of a question from its execution
- Neural vs. Symbolic
- Our Proposal: coupling the two views
- Conclusion

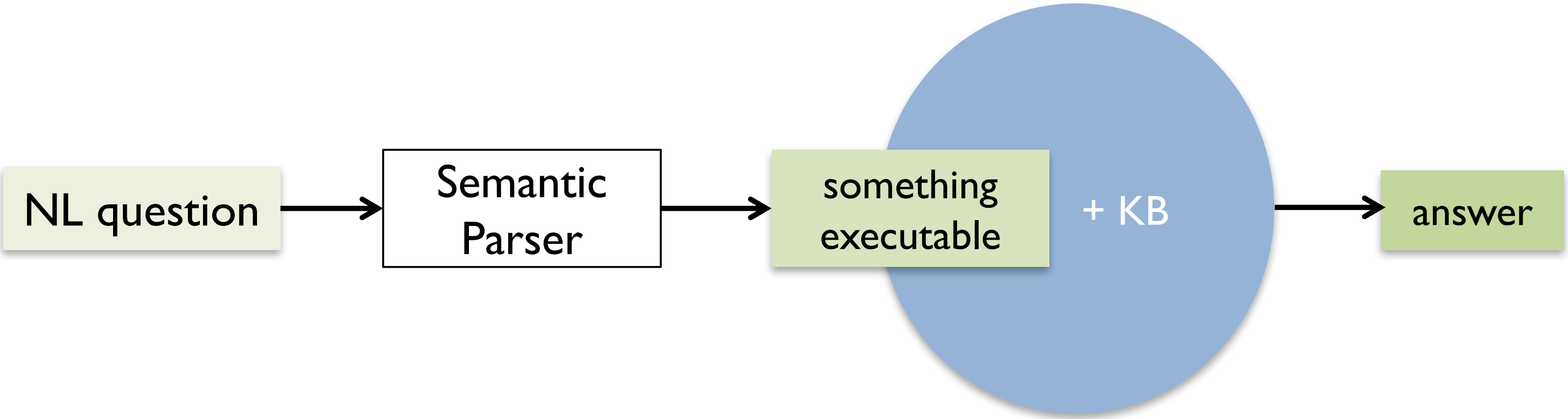
Outline

- Learning the semantics of a question from its execution
- Neural vs. Symbolic
- Our Proposal: coupling the two views
- Conclusion

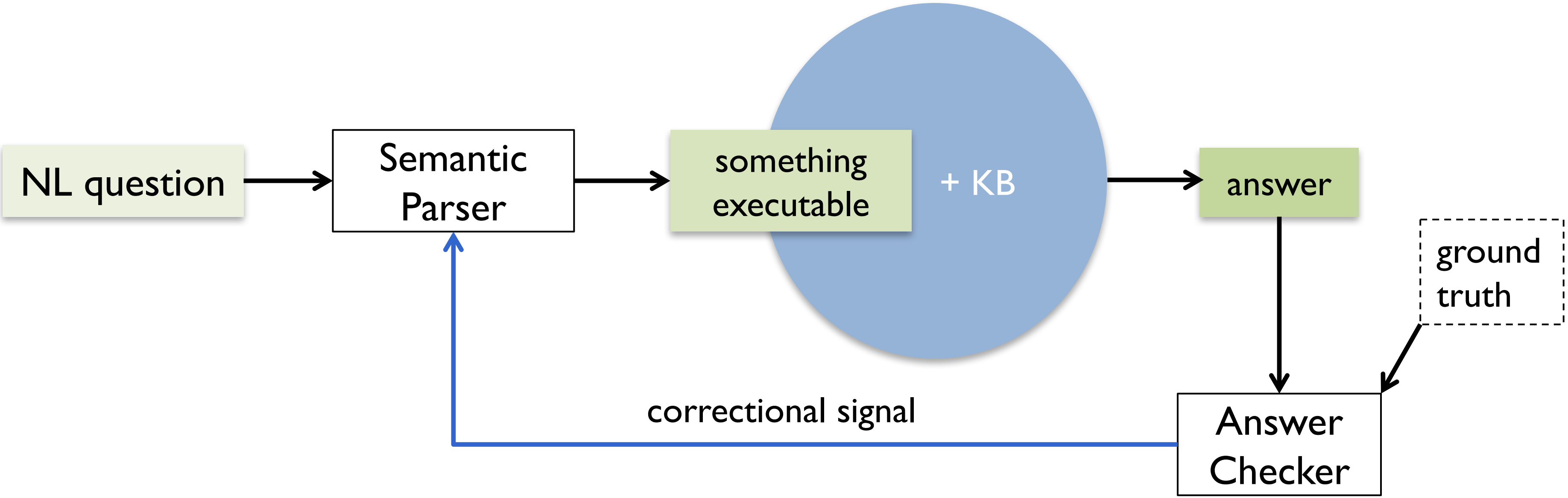
Learning the semantics of a question from its execution



Learning the semantics of a question from its execution



Learning the semantics of a question from its execution



Two parsing choices: (1)

① Question into a composite SQL-like command (Liang et al. ACL-17)

Query:

How long is the game with the largest host country size?

Knowledge base (table):

Year	City	···	Area	···	Duration
···					
2000	Sydney	···	200	···	30
2004	Athens	···	250	···	20
2008	Beijing	···	350	···	25
2012	London	···	300	···	35
2016	Rio de Janeiro	···	200	···	40
···					

Semantic parsing →

```
select Duration where
                  area = max(area)
```

It is essentially a sequence-to-sequence model, while the output sequence is executable

Two parsing choices: (2)

① Question into a composite SQL-like command (Liang et al. ACL-17)

② Question into a sequence of "primitive" operations (Neelakantan et al. ICLR-16, Yin et al. IJCAI-16)

Query:

How long is the game with the largest host country size?

Knowledge base (table):

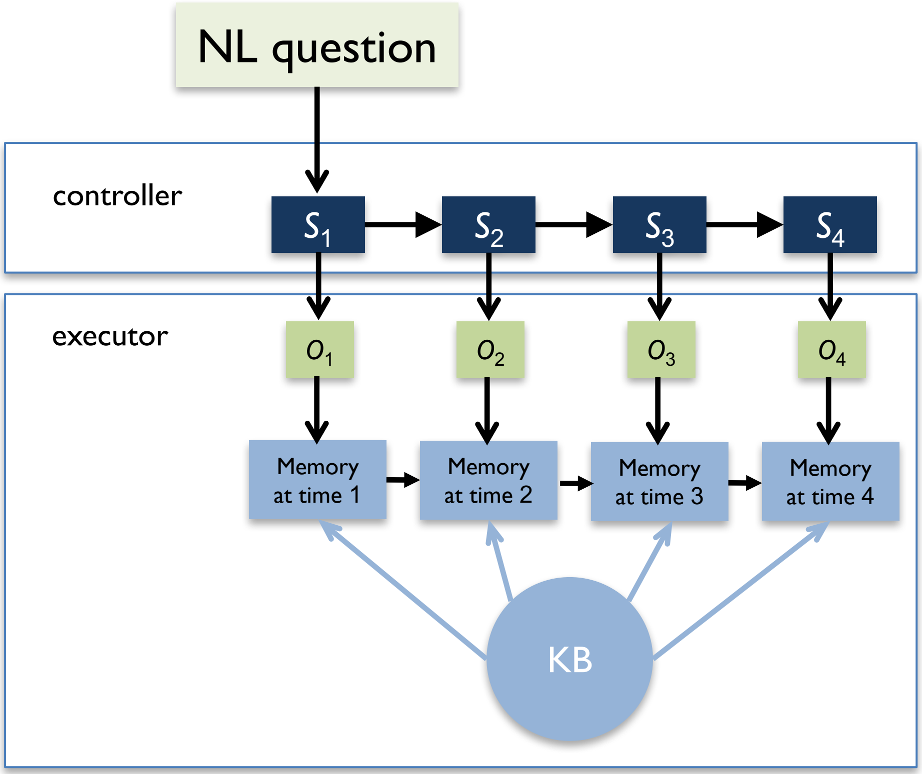
Year	City	···	Area	···	Duration
···					
2000	Sydney	···	200	···	30
2004	Athens	···	250	···	20
2008	Beijing	···	350	···	25
2012	London	···	300	···	35
2016	Rio de Janeiro	···	200	···	40
···					

Semantic parsing

STEP-1: Row selection:
`argmax(area)`

STEP-2: Value selection:
`select_value(Duration)`

Question as a sequence of operations



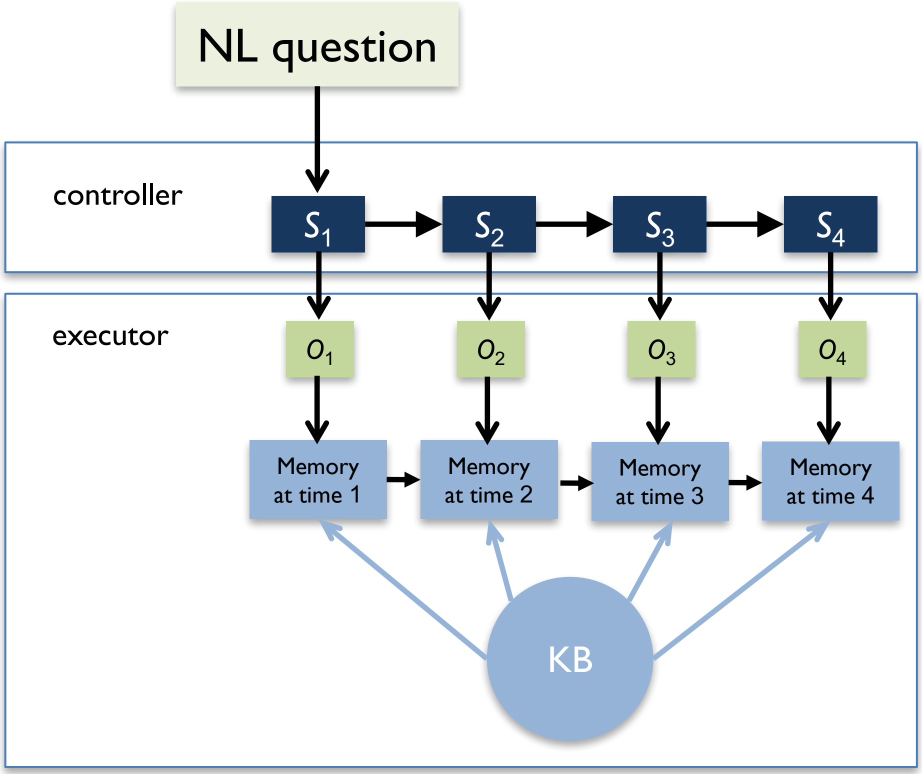
1. Neural net controller sequentially emits operations (with argument)
2. Each operation has its own semantics, so the operation of each step can be potentially supervised
3. Each operation is applied on KB and memory from previous operation, with the new result saved in the current memory

We will use this as our base models

Outline

- Learning the semantics of a question from its execution
- Neural vs. Symbolic
- Our Proposal: coupling the two worlds
- Conclusion

We choose to parse questions into a sequence of operations



Symbolic executor vs. Neural executor

Again we have two modeling choices

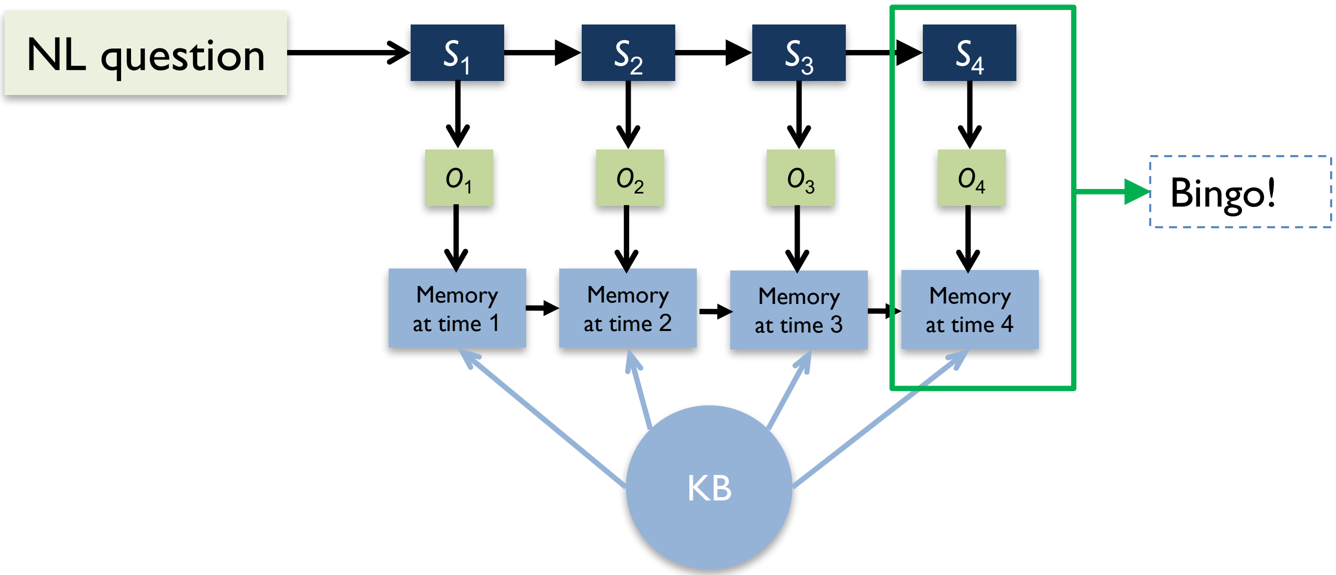
① **Symbolic Executor:** the execution is purely symbolic, while the controller is neural net-based, whose optimization objective is **non-differentiable**

② **Neural Executor:** everything is "neuralized", including the executor and the intermediate memory, so the objective is naturally **differentiable**. Although it is easy to learn, it suffers from low execution efficiency and low generalization ability

The choice of Neural Programmer (Neelakantan et al. ICLR-16) is an interesting middle course, but we don't consider it due its limited potential for complex operations

Choice-I: Symbolic executor

- Learning is hard (with reinforcement learning):
  - relatively big action space: primitive operators x argument
  - only final reward (when the executions return the correct result)



Examples of symbolic operators

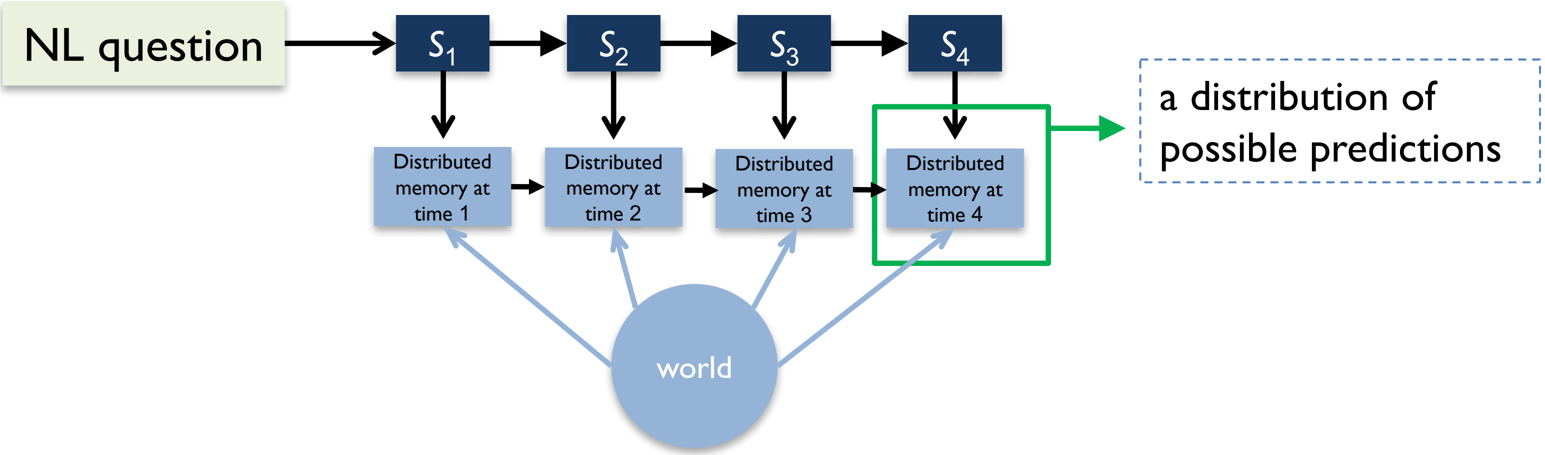
- We limit ourselves with knowledge-base with a single table
- Each execution on a table is specified by a primitive operator with an argument

  Example: `argmax(year)` selects the row with the field `year` having the greatest value

Operator	Explanation
`select_row`	Choose a row whose value of a particular column is mentioned in the query
`argmin`	Choose the row from previously selected candidate rows with the minimum value in a particular column
`argmin`	Choose the row from previously selected candidate rows with the maximum value in a particular column
`greater_than`	Choose rows whose value in a particular column is greater than a previously selected row
`less_than`	Choose rows whose value in a particular column is less than a previously selected row
`select_value`	Choose the value of a particular column and of the previously selected row
`EOE`	Terminate, indicating the end of execution

Choice II: Neural executor

- Neural Enquirer (Yin et al. IJCAI-16) as the example: Learning is typically easy through normal back-propagation. It can learn to deal with quite complicated questions
- Its execution efficiency is low due to its fully neural architecture, and the accuracy on parsing complex questions is not satisfying

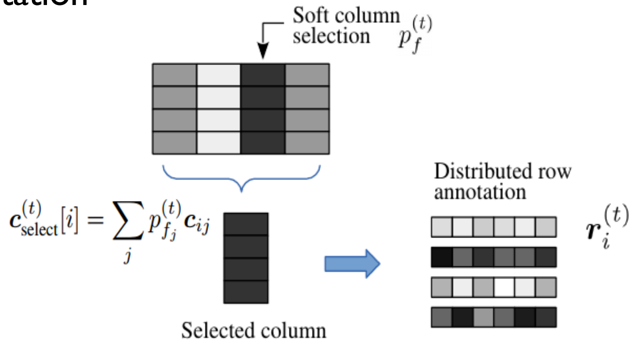


Neural Enquirer: Overall diagram

- Embed the table: keep the table structure, but embed the value and field
- Fully "neuralized" execution (matrix/vector processing with gating and pooling)
- Stacked layers of (Excutor, Memory) pairs to mimic the sequence of operations, while the memory saves the intermediate result of each layer of execution

Each execution step in Neural Enquirer includes

- Soft column attention (**this part is naturally interpretable**)
- Distributed row annotation



Neural vs. Symbolic

	Symbolic	Neural	Wanted
Learning Efficiency	Very low	High	High
Execution efficiency	High	Low	High
Interpretability	High	Low	High
Accuracy	Low	Low	High

Neural vs. Symbolic

	Symbolic	Neural	Wanted
Learning Efficiency	Very low	High	High
Execution efficiency	High	Low	High
Interpretability	High	Low	High
Accuracy	Low	Low	High

Can we (sort-of) have the best of both worlds?

Outline

- Learning the semantics of a question from its execution
- Neural vs. Symbolic
- Our Proposal: coupling the two views
- Conclusion

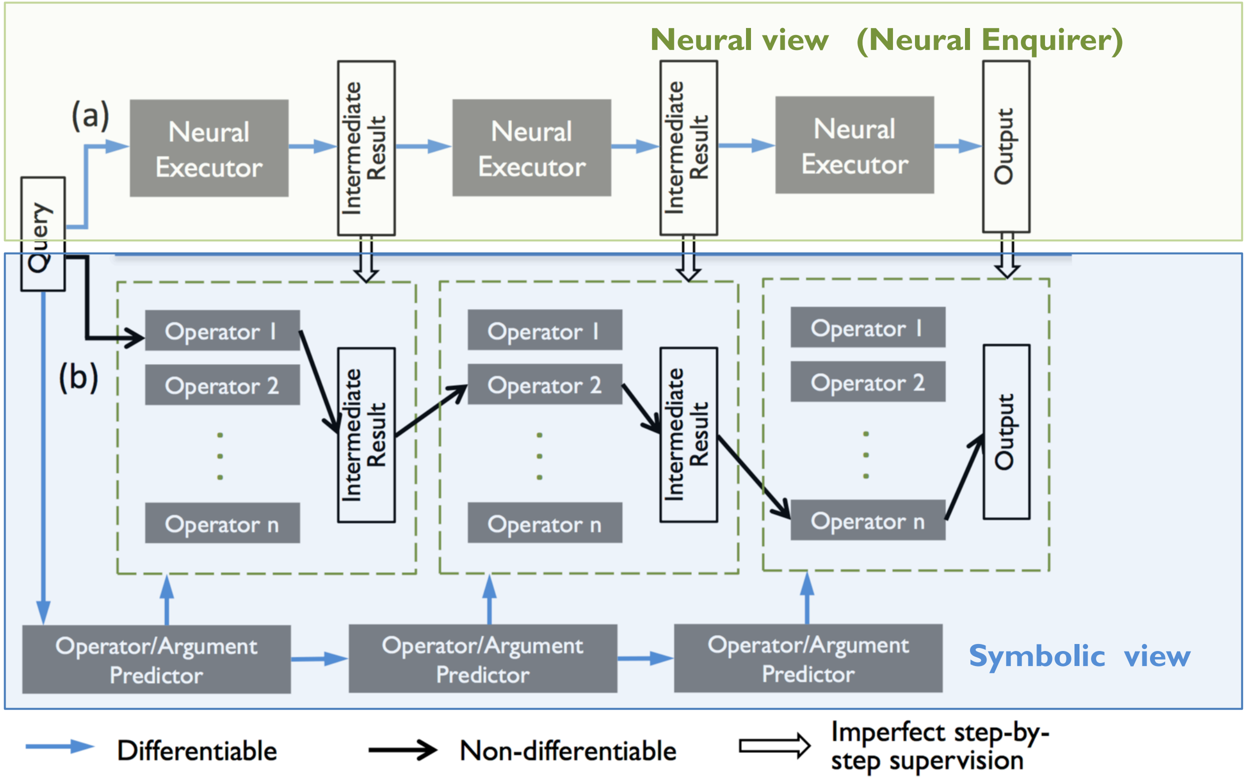
General intuition

- Neural models and symbolic models are like two different views of the same complex semantic parsing process

- We can maintain both views in the same system, and let them talk to each other, to encourage some consistency between the two views

- It is a bit like Multi-view Learning, while in this work the contrast of views come from **intrinsic representation choices**, instead from different **given** aspects or features of the same object

The diagram



1. We have both neural and symbolic view in the same system
2. There is information exchange between the two views during the training
3. We will use only symbolic view for testing after the training is done, for high execution efficiency

Coupling the two views

General idea: (distributed ⟶ symbolic)

- **STEP-1:** Train the neural model as in (Yin et al. IJCAI-16) in an end-to-end fashion
- **STEP-2:** Pre-train the field selection part of the symbolic model with the prediction of the neural model trained in STEP-1 in a step-by-step way
- **STEP-3:** Train the symbolic model with REINFORCE with the execution accuracy as reward

Coupling the two views

General idea: (distributed $\longrightarrow$ symbolic)

- **STEP-1:** Train the neural model as in (Yin et al. IJCAI-16) in an end-to-end fashion
- **STEP-2:** Pre-train the field selection part of the symbolic model with the prediction of the neural model trained in STEP-1 in a step-by-step way
- **STEP-3:** Train the symbolic model with REINFORCE with the execution accuracy as reward

(distributed $\longrightarrow$ symbolic $\longrightarrow$ distributed)

- **STEP-4: (Feedback step)** Use the symbolic model to train the attention of the neural model in a step-by-step way

Coupling the two views

General idea: (distributed $\longrightarrow$ symbolic)

- **STEP-1:** Train the neural model as in (Yin et al. IJCAI-16) in an end-to-end fashion
- **STEP-2:** Pre-train the field selection part of the symbolic model with the prediction of the neural model trained in STEP-1 in a step-by-step way
- **STEP-3:** Train the symbolic model with REINFORCE with the execution accuracy as reward

(distributed $\longrightarrow$ symbolic $\longrightarrow$ distributed)

- **STEP-4: (Feedback step)** Use the symbolic model to train the attention of the neural model in a step-by-step way

STEP-2 and **STEP-4** approximately maintain the consistency of the two views on field selection

Pre-training with supervision form neural view

- Let m be the number of actions to pre-train, J is the function to be maximize
- Only the parameters associated with field selection is trained in this phrase, the other parts are left dangling

Imperfect supervision signal from
Neural Enquirer MAP prediction

$$J = -\sum_{i=1}^{m} \sum_{j=1}^{n_{\text{label}}^{(i)}} \hat{t}_j^{(i)} \log p_j^{(i)}$$

Step-by-step supervision

- We used supervised learning for pre-training, but many other ways (eg, some smart sampling) may also work

Policy improvement with REINFORCE

- $$J = -\mathbb{E}_{a_1, a_2, \cdots, a_n \sim \theta}[R(a_1, a_2, \cdots, a_n)]$$
- Gradient: $$\frac{\partial J}{\partial \boldsymbol{o}_i} = \tilde{R} \cdot (\boldsymbol{p}_i - \mathbf{1}_{a_i})$$
- Reward *R* : 1 for correct result, 0 otherwise
- Tricks
  - Exploring with a small probability (0.1)
  - Subtracting the mean (reinforcement comparison)
  - Truncate negative reward (reward-inaction

Experimental setting

- Dataset: from (Yin et al. IJCAI-16)
  - Synthesized data: table has 10 fields (columns) and 10 rows, about Olympic games

Query:

How long is the game with the largest host country size?

Knowledge base (table):

Year	City	···	Area	···	Duration
···					
2000	Sydney	···	200	···	30
2004	Athens	···	250	···	20
2008	Beijing	···	350	···	25
2012	London	···	300	···	35
2016	Rio de Janeiro	···	200	···	40
···					

  - 25k samples (different queries and tables): tables are randomly generated, the questions and answers are generated accordingly.
  - Many questions are extremely complicated, eg `"How long is the last game which has smaller country size than the game whose host country GDP is 250?"`

Experimental results: Accuracy

Query type	SEMPRE†	Denotation			Execution		
		Distributed†	Symbolic	Coupled	Distributed	Symbolic	Coupled
`SelectWhere`	93.8	96.2	99.2	**99.6**	–	99.1	**99.6**
`Superlative`	97.8	98.9	**100.0**	**100.0**	–	**100.0**	**100.0**
`WhereSuperlative`	34.8	80.4	51.9	**99.9**	–	0.0	**91.0**
`NestQuery`	34.4	60.5	52.5	**100.0**	–	0.0	**100.0**
Overall	65.2	84.0	75.8	**99.8**	–	49.5	**97.6**

Pasupat & Liang, ACL-16
Compositional semantic parsing on semi-structured tables.

Experimental results: Accuracy

Query type	SEMPRE†	Denotation			Execution		
		Distributed†	Symbolic	Coupled	Distributed	Symbolic	Coupled
`SelectWhere`	93.8	96.2	99.2	**99.6**	–	99.1	**99.6**
`Superlative`	97.8	98.9	**100.0**	**100.0**	–	**100.0**	**100.0**
`WhereSuperlative`	34.8	80.4	51.9	**99.9**	–	0.0	**91.0**
`NestQuery`	34.4	60.5	52.5	**100.0**	–	0.0	**100.0**
Overall	65.2	84.0	75.8	**99.8**	–	49.5	**97.6**

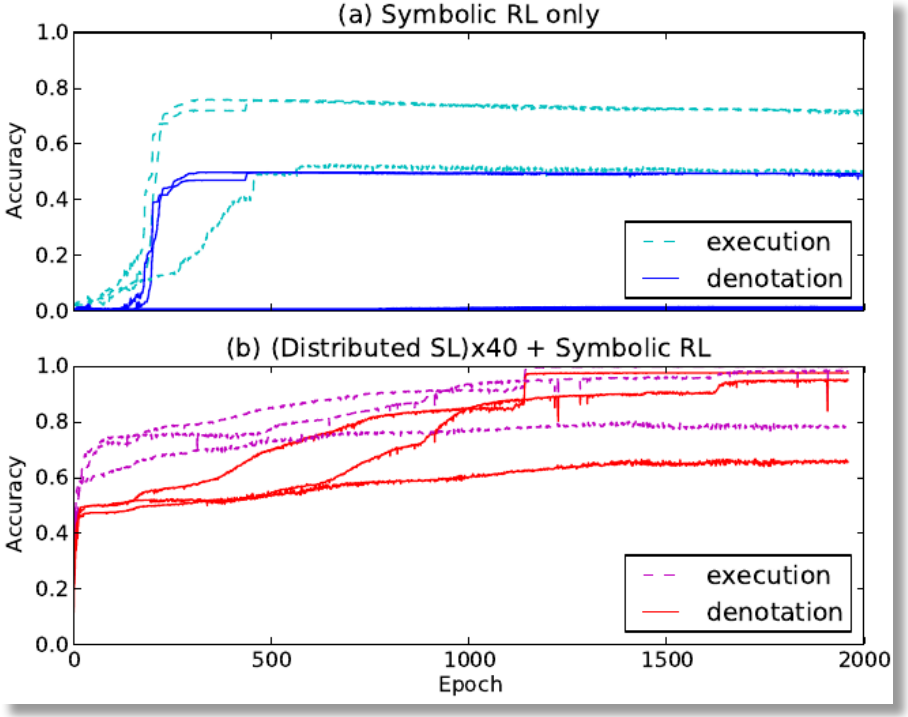
accuracy on giving
the right answer

Experimental results: Accuracy

Query type	SEMPRE[†]	Denotation Distributed[†]	Denotation Symbolic	Denotation Coupled	Execution Distributed	Execution Symbolic	Execution Coupled
SelectWhere	93.8	96.2	99.2	**99.6**	–	99.1	**99.6**
Superlative	97.8	98.9	**100.0**	**100.0**	–	**100.0**	**100.0**
WhereSuperlative	34.8	80.4	51.9	**99.9**	–	0.0	**91.0**
NestQuery	34.4	60.5	52.5	**100.0**	–	0.0	**100.0**
Overall	65.2	84.0	75.8	**99.8**	–	49.5	**97.6**

accuracy on being right
on every execution

Experimental results: Learning efficiency



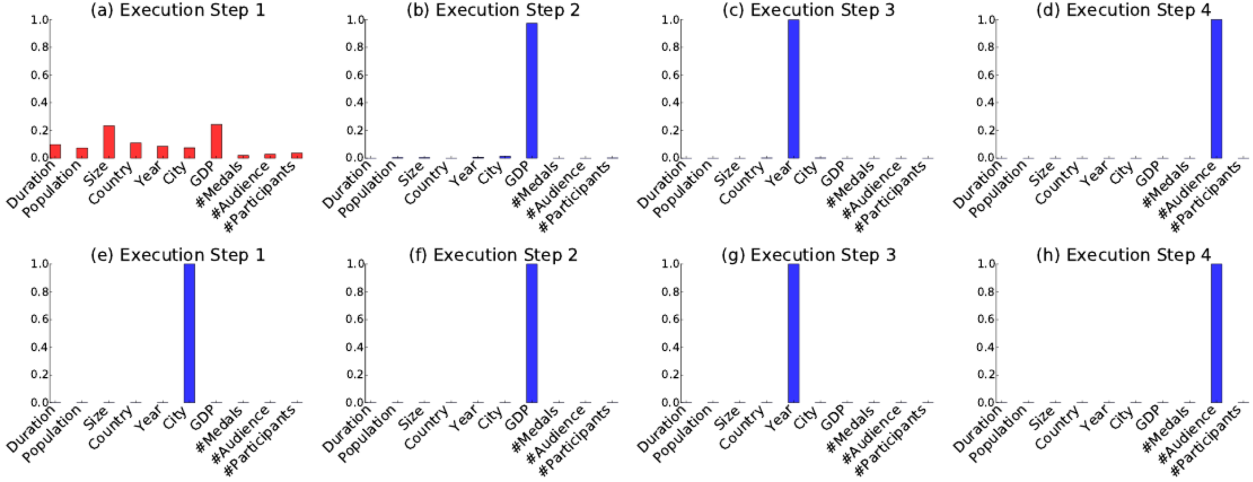
Experimental results: Execution efficiency

	Fully Distributed	**Our approach**		
		Op/Arg Pred.	Symbolic Exe.†	Total
CPU	13.86	2.65	0.002	2.65
GPU	1.05	0.44		0.44

Experimental results: with feeding-back

Training Method	Accuracy (%)
End-to-end (w/ denotation labels)†	84.0
Step-by-step (w/ execution labels)†	96.4
Feeding back	96.5

Query: How many people watched the earliest game whose host country GDP is larger than the game in Cape Town?



Outline

- Learning the semantics of a question from its execution
- Neural vs. Symbolic
- Our Proposal: coupling the two views
- Conclusion

Conclusion and future work

- Coupling the symbolic view and distributed view in one model might be better than either one working alone, especially on hard problems
- We are looking for broader more profound ways to combine symbolic model and neural models in real-world semantic parsing tasks

Thank you

Poster #36　（today）

for more questions and job positions at DeeplyCurious